Coral Damage Assessment and Restoration Tools for Small Vessel Groundings (Extension of Seagrass Mini-312 Program to Coral Reefs)

Project Report to NOAA Damage Assessment Center, Silver Spring, Maryland

by

Paula E. Whitfield, Mark S. Fonseca, W. Judson Kenworthy
NOAA, National Ocean Service
National Centers for Coastal Ocean Science
Center for Coastal Fisheries and Habitat Research
101 Pivers Island Road
Beaufort, North Carolina 28516-9722

November 1, 2001

paula.whitfield@noaa.gov , mark.fonseca@noaa.gov, jud.kenworthy@noaa.gov

Executive Summary

Injury to NOAA trust resources requires an accurate assessment of the time it will take to recover to pre-injury conditions. Previous work has demonstrated that the geometry of an injury greatly influences the outcome of that assessment. Therefore, in order to determine accurately this recovery, we have developed spatial modeling processes to perform these computations that can accommodate geographically accurate data collected from the injury site. This work was conducted using two fundamentally different software systems; a probabilistic technique using SAS7 and a deterministic approach using ArcINFO7. The results of this work has produced two comparable spatial modeling processes that deal not only with any injury geometry, but have sufficient flexibility to insert a wide variety of recolonization modes (including both asexual and sexual processes, such as rhizome extension, or seed, spore and larval recruitment). The output of these models is used to provide a mathematical formula that can be used to directly compute lost interim resource services, a key element in NOAA's damage assessment protocol. Moreover, the approach developed here should be directly applicable to virtually any sessile community under NOAA's stewardship.

Introduction

Significant progress has been made over the last several years in our ability to scale up information regarding the recovery of seagrass injuries, from the individual plant level to that of motor vessel groundings. Our earlier modeling efforts employed a simple programming language (STELLA7) that allowed us to incorporate simple geometric aspects of plant spreading dynamics across the seafloor. That model was based on explicit empirical information including rhizome extension rates, rhizome branching frequency and rhizome branching angles to make predictions of the overall spreading rate and was used successfully by NOAA in federal court claims cases. However, this approach was cumbersome to adapt to the wide range of injury geometry and recolonization strategies that strongly influence the rate at which recovery occurs for the various NOAA trust resources (e.g., marshes, corals, mangroves). Because the injury recovery rate was critical to the process of injury assessment by DARP (Fonseca et al. 2000), it was imperative that predictions of resource recovery be responsive to injury geometry and intrinsic recolonization dynamics.

To meet this challenge, we have developed a spatial modeling process that deals not only with any injury geometry, but has sufficient flexibility to insert a wide variety of recolonization modes (including both asexual and sexual processes, such as rhizome extension, or seed, spore and larval recruitment). The key to the flexibility of our modeling approach was to collapse three and four-dimensional processes into a simple two or three-dimensional representations by explicitly modeling recovery of a resource in space and time (i.e. 100% recovery to pre-injury baseline conditions). In order to fully evaluate the efficacy of this approach by the DARP, we have employed two very different, but parallel techniques using two fundamentally different software systems; a probabilistic technique using SAS7 and a deterministic approach using ArcINFO7.

Due to severe illness, the member of our team that was leading the SAS-based modeling technique was forced to retire from Federal service. Having recovered the models from his computer, we are only now beginning to interpret the intricacies of his modeling approach in an attempt to continue this line of experimental model development. Because of that setback, we cannot report as much progress in that area as had been planned. However, we feel that we can rapidly make up this ground early in FY 02, providing us with important comparative capabilities and cross-checking that are inimical to model development. Despite this setback, there has been significant progress in the translation of the original seagrass-based model to a form that was suitable for application to any trust resource.

Objectives

The primary goal of this project was to adapt the existing seagrass habitat recovery models (Fonseca et al. In prep.) developed for the mini-312 protocol to coral reef injuries. To meet this goal, we created a model capable of incorporating current biological information on coral reef recovery from the existing literature, and create a spatially and temporally accurate representation of the coral recovery process following a vessel grounding. There were three specific objectives;

1. Adapt the mini-312 seagrass Arc/INFO-based spatial recovery model to a coral ecosystem. The model structure would include coral propagule establishment and vegetative growth

2. Develop a conceptual framework for inclusion of a third spatial dimension in the coral reef recovery model; this entails modeling specific growth of corals as opposed to the original mini-312 model metric which was establishment of 100% recovery in each unit area of the modeling universe.

3. Adapt the mini-312 SAS based recovery model to coral recovery and compare the SAS-based versus the Arc/INFO-based spatial model.

Seagrass Recovery Model Structure - Arc/INFO ^{tm}

The model structure for coral recovery was adapted from the existing seagrass habitat recovery model. Seagrass recovery was examined using a grid-based approach developed in Arc/INFO. The injury perimeter was modeled as a single moving front of 100% seagrass cover (Fonseca et al. In prep.) by combining a measure of Euclidean geometry with a weighted cost-distance function to calculate the shortest distance and lowest cost from each cell representing the injured area to the closest, uninjured cells in the grid. This method has been used to examine squirrel dispersion in Canada (Gonzales 2000), predict amphibian and reptile dispersion (Strager et al. 2000), predict the migratory pathways of mink and warblers (Halpin and Bunn 2000) and to examine the connectivity of habitats for larval fish in a marine environment (Halpin and Bunn 2000).

Specifically, the seagrass model was an iterative process that begins with the Euclidean distance function (**Eucdist**) in the Arc/INFO GRID module. The flowchart in Figure 1 outlines this process. The first grid created (source grid) was input into the **Eucdist** function to output a grid whereby every non-source cell was designated a value that represented the closest distance of each non-source cell to the closest source cell (Figure 1c). The next step in the process utilized the **Cost Distance** function, incorporating both the original source grid and the output grid from the **Eucdist** function to create a second output grid of values that represent cost (Figure 1d). Each cell value in this grid represents the cost of moving through each cell in cost per unit distance. This **Cost Distance** function is a weighted distance function that incorporates both the distance and direction of each non-source cell to the closest source cells. This weighted distance function is based on node link graph theory (ESRI, Arc/INFO v8.0.1). This new **Cost Grid** can be in any unit of cost that are being measured such as time, dollars or preference. In this case, the **Cost Grid** was actually representing time to recovery since it was based on distance and can be conceptually equated to a physiological cost associated with vegetative colonization of new space. The greater the distance of the non-source cell to the source cells, the longer it will take for the seagrass (source) to colonize this area. The cells in the **Cost Grid** that represent the lowest cost values are reclassified as ‘source cells’ and merged with the original source grid, creating a ‘new source grid’ (Figure 1e). At this point the whole process, beginning with the **Eucdist** function, starts over (Figure 1b) with the new source grid and continues iteratively until all of the cells are designated as being filled with seagrass. At each iteration (time step) the number of cells filled are output into an ASCII text file (Figure 1f). Output from each time step of the model is the percent recovery and, upon completion of the model, the number of years to recovery at each time step is calculated.

In the present form of the model, the effects of the **Eucdist** function and **Cost Distance** function are somewhat similar. **Eucdist**, however, considers only distance to the nearest source cells (those in a colonized state) for each iteration of the model. Similarly**, Cost Distance** implicitly considers the distance from each non-colonized (non-source) cell to the closest source cells and assigns a cost (based on the cost grid) of passing from cell to cell. Therefore, the fundamental difference between the **Eucdist** function and **Cost Distance** function is that **Cost Distance** not only considers the effects of source cells in all directions, it also can assign a cost for passing through intervening cells. This leaves the programmer an opportunity to embed directionally dependent information that can promote or retard the contribution of other cells (e.g., seafloor slope and composition, wave direction). In this case, the output grid from the **Eucdist** function is the only additional information we are supplying to influence the cost of moving between cells. We are not currently embedding any other ancillary data such as seafloor slope, but experiments are underway that should yield empirical evidence for addition of these spatially articulated, controlling factors.

Our choice of scale for conducting this work is reflected by the grid cell size. The cell size was 10 x 10 cm. However, to conform to Standard International units, the number of iterations output by the model represents a distance converted to meters. This was the value that was used to calculate years to recovery. Based on published values (see Fonseca et al. In prep.) and new, empirical information generated under our research program (Kenworthy et al. In press) the rate of seagrass growth in years / meter are used in the following equations:

Percent recovery = #. of cells filled at each time step/ total #. of injured (i.e., non-source cells) (1)

Years to recovery = # of iterations * rate of growth (years m^{-1}) (2)

These two variables, percent recovery and years to recovery, are then exported to a text file, incorporated into SAS^{tm} , and plotted to derive a recovery function (typically a quadratic equation) that describes the recovery horizon for the entire injury (Figure 1f).

Seagrass Recovery Model Structure - SAS^{tm}

With this approach, we created a simulation of the invasion of an injury scar through lateral extension of rhizomes and shoots of seagrass surrounding the injury site. We began by invoking Proc IML, the interactive matrix language in SAS. The value and utility of Proc IML was that it allowed us to readily address each element of a matrix, where each element of the matrix represents a pixel in the simulation (injury) space. For example, we can ask whether a particular pixel, $\mathbf{P}_{ij,}$ at time t=t' is vegetated (i.e. whether the corresponding matrix element has a value of 1) or unvegetated (has a value of 0). We next used Proc IML to create an n X n matrix, **P**, with each element set to zero. The next step was to create the initial conditions for the simulation. This meant surrounding the hypothetical injury with vegetated pixels by setting appropriate elements of **P** to a value of 1. A second n X n matrix, $\mathbf{E}_\mathbf{d}$, was then created for the purpose of keeping track of the number of edges of each pixel in **P** that are on a vegetated-unvegetated boundary. Initially, elements of $\mathbf{E}_\mathbf{d}\mathbf{E}$ can only be zero or one, because a pixel can only be on the edge of the injury or not. However, once the simulation of recovery, moving laterally into the injury gets underway, an element of $\mathbf{E}_\mathbf{d}$ can range from zero to three. If we allowed the simulation to include revegetation of pixels within the injury by random seeding events, then an element of $\mathbf{E}_\mathbf{d}$ might be as high as four because a seeded pixel might be completely surrounded by unvegetated pixels.

The next step was to initiate a series of repetitive computations by invoking a user-defined module in Proc IML (named "RECOVERY"). During each successive time step, RECOVERY computed the elements of $\mathbf{E}_\mathbf{d}$ for all pixels within the original injury site. That is, it counted the number of boundary edges for each pixel. It then examined each pixel within the original injury (i.e. each element within the interior of **P**). Each pixel within **P** (value = 0) was examined to determine whether there was a vegetated border pixel. If so, then the open pixel was converted into a filled, or colonized state (value = 1). Thus, the more irregular the shape of the injury boundary, the more pixels are converted to "filled" per iteration of the module. Before the time step was incremented and the whole process repeated, the elements of **P** and $\mathbf{E}_\mathbf{d}$ were copied into SAS data sets for later use in plotting various aspects of the sequential recovery process, also allowing one to watch the invasion process advance with the successive iterations.

Adaptations of the *Seagrass Recovery Model* to Corals using Arc/INFO:

In contrast to seagrass recovery, which is strongly influenced by vegetative growth from adjacent, uninjured plants along the perimeter of the injury(source cells), coral reef recovery is mainly due to propagule recruitment by several different species (Connell 1997). The **Seagrass Recovery Model,** described above was modified to include different growth rates and propagule recruitment events for three different coral species. The structure of this model allows concurrent simulation of as many coral species (or other taxa and mixtures of taxa) as necessary.

As before, the initial grid is the shape and size of the injury perimeter, at a resolution of 10 x 10 cm. For purposes of internal differentiation by the software, all cells outside the perimeter will be set to values of 99 (any value can designate a source cell, therefore this value is not ordinal) and inside the perimeter the cells are either "no data" or designated as propagules (Figure 2a) as the result of a random selection process. Propagules are set to values of 2, 3 and 4 to represent 3 different species. These initial propagules were chosen by selecting random coordinates from within the injured area (Figure 2b). For model simulation purposes, horizontal growth rates of 10, 30 and 50 cm yr^{-1} were chosen; growth rates considered to be realistic for several species of colonizing Western Atlantic corals (Zengel and Hinkeldey, 2001).

Unlike the **Seagrass Recovery Model**, the perimeter cells (also called the initial perimeter source cells) were used to establish the boundary of the injury. Ingress of growth from the perimeter was not modeled in this example because horizontal expansion by corals is widely considered to contribute little to the colonization of large, open spaces (Connell 1997). Therefore, only the randomly inserted propagules were allowed to grow and contribute to recovery. However, the method described below can also be used for including perimeter growth if that is considered necessary. To meet objectives 1 and 2 (above), the methods for horizontal growth and vertical growth of established propagules will be presented separately in the next two sections.

Modeling Horizonal Growth: The original grid (Figure 2b.) contains three possible sources of information that can contribute to colonization of uncolonized cells. In this example, the perimeter is not considered a source, but is used as a barrier to propagule growth and recruitment. The first step in the model was to run the **Eucdist** function on the original grid. The output of the **Eucdist** function was input to the **Cost Allocation** function to create a **Cost Grid** and a **Cost Allocation Grid**. The least accumulative cost values in the **Cost Grid** that is created during the previous step is examined to determine the lowest cost of growing horizontally10 cm/yr, 30 cm/yr, and 50 cm/yr respectively (Figure 2c). Figure 2c is an example of the different cost values that are associated with the three growth rates. The **Cost Allocation Grid** defines for each cell the source that could be reached by the least accumulative cost. Figure 2d is an example of the **Cost Allocation Grid** where the different colors identify the closest source cells that can be reached with the least accumulative cost. This grid is important for defining analysis masks. Analysis masks are used to limit an analysis to a specific part of a grid. In this case, the analysis of each species can be isolated by assigning the analysis mask to one particular species at a time. Figure 2d-e illustrate how analysis masks are used to modify the **cost grid** by species. The analysis mask is set for each species and the **Cost Grid** for each species is modified in order to apply differential growth rates. Without using an analysis mask there would be no way to modify the cost associated with the different species separately. In our example, the **Cost Grid** for source 2 (species) is not modified at all. The **Cost Grid** for source 3 was divided by 4.5, and the **Cost Grid** for source 4 was divided by 12. These values were chosen to generate faster growth rates for species 3 and 4 respectively, with species 2 growing the slowest. Therefore, the faster the species grows the lower the cost values, as cost represents an impediment to growth.

Once the **Cost Grid**s for each species are modified (Figure 2d) they are merged. creating the final **Cost Grid** (Figure 2e). With the creation of the final **Cost Grid**, the methods for horizontal growth and vertical growth diverge. To model horizontal growth, the cells with lowest values generated during each iteration of the model are selected and turned back into their respective source values (Figure 2g).

Propagule recruitment then occurs by randomly selecting x and y coordinates from within the non-colonized area are chosen. These new recruits are also assigned the values of 2, 3 or 4, in equal proportion among species (variable recruitment can also be included). After the random recruitment event, two additional steps were taken. First, the number of colonized cells and their values were output into an ASCII text file for further analysis (see equation 1 and 2, Figure 2i). Second, this grid was used as the new source grid (Figure 2h), whereupon, the whole process starts over again, building on the existing growth of old propagules and the new ones (Figure 2a). At model completion, the percent recovery and years to recovery are plotted in SAS for each species to derive the recovery horizon (Figure 3). Output from each iteration of the model was scaled *post facto* to represent one year.

Modeling Vertical Growth: In each iteration after the individual **Cost Grid**s are merged (Figure 2d) the lowest values are selected and turned into values of 1 (Figure 2e) creating the **Age Grid**, a counting mechanism that records the number of model iterations to provide age input to the colonized cells. To accomplish this, the **Age Grid** is saved and summed to each new **Age Grid** that is created so that the highest value in the grid represents the number of iterations (Figure 2e). An example of an **Age Grid** after 16 iterations is shown in Figure 2e. Even though the coral propagule growth has filled in the horizontal area, the coral propagules can continue to 'age'. A two and three-dimensional illustration of the **Age Grid** are shown in (Figure 3). The bottom panel is a three-dimensional representation of the coral surface after all cells have been colonized, but vertically scaled based on the **Age Grid**. Vertical growth rate data could also be applied at each iteration as a multiplier of the **Age Grid** for each species, to obtain an accurate spatial representation of vertical growth. This methodology represents a straightforward conceptual template for modeling three-dimensional growth with the data that are presently available.

Example Recovery Horizons: As mentioned above, each model iteration outputs the sum total of occupied cells to an external (ASCII) file. Given the spatial resolution of the model and knowing the horizontal spreading rate, the relative growth rates used in the independently modified Cost Grids can be used to time scale the iteration as the inverse of horizontal growth velocity (yielding years m^{-1}). Therefore, the change in the number of colonized cells between iterations as compared to the time 0 number of cells yields a percent of the original injury that has been colonized, scaled with respect to time (Figure 4). Vertical growth can be modeled in a similar fashion, so long as a reliable estimate is obtained regarding the final dimensions of the coral structure for a given coral species that is being "aged" with the **Age Grid** function in each colonized cell.

Adaptations of the *Seagrass Recovery Model* to Corals using SAS:

In this adaptation of the seagrass-based SAS model, we created a simulation of the invasion of a coral grounding through planktonic coral propagules. As before, we began by invoking Proc IML, the interactive matrix language in SAS. Once an initial random seed was chosen, and a propagule rain has occurred, we again query whether a particular pixel (here a square meter of uncolonized substrate), $\mathbf{P}_{ij}$, at time t=t' is colonized (i.e. whether the corresponding matrix element has a value of 1 or 0). The next step, creating an edge-tracking matrix was the same as with the seagrass version.

However, once the simulation of growth out from the random propagule has begun moving laterally into the injury, an element of $\mathbf{E_d}$ can range from zero to four because a seeded pixel might be completely surrounded by uncolonized pixels.

The next step was to again initiate a series of repetitive computations by invoking a user-defined module in Proc IML (named "**CORAL**"), which performed the comparison to determine whether there was a colonized border pixel. If so, then the open pixel was given a random number between 0 and 1. Given our empirical information from the literature, we then applied a Boolean test wherein the pixel would be converted into a filled, or colonized state (value = 1) if the random value were above some value (e.g., 0.85 – a 15% chance of colonization with that iteration). By varying this probability, we can control the rate of lateral occupation of space by a propagule once it has set. Moreover, the more irregular the shape of the injury boundary, the more pixels that have a chance of being evaluated for conversion to being colonized per iteration of the module. Before the time step was incremented and the whole process repeated, the elements of **P** and $\mathbf{E_d}$ were copied into SAS data sets for later use in plotting various aspects of the sequential recovery process, also allowing one to watch the invasion process advance with the successive iterations. Thus, by setting the threshold probability function and *post facto* scaling of model iterations to time, we have dual controls on the recolonization process. An example of a time 0 propagule set and its spread through 99 iterations of the model is given in Figure 5.

Modeling Assumptions

These models assume no mortality, however the density of the propagules can be scaled to represent survival. Moreover, there are no effects of disturbance events (storms) that are extremely likely to occur in the protracted time period that is intrinsic to coral reef recovery. With this assumption, the Responsible Party enjoys a utopian injury recovery scenario. Although not always the case in nature, a radial growth pattern has been employed for all propagules as we are not aware of any empirical data that would allow us to reformulate the model with anisotropic growth. Also, based on existing literature, propagules will only recruit into disturbed areas without existing propagules. They will not recruit into areas already colonized. Once a cell is colonized, it cannot receive additional colonization. Therefore, given a fixed recolonization rate and a declining number of cells that are available for colonization, the colonization rate increases geometrically, another significant benefit for the Responsible Party. A feedback loop that scales random recolonization to available space is under consideration. Finally, it needs to be noted that variable recruitment flux has not been included at this point. However, by setting recruitment rate to an average value, this acts to even out the influence of recruitment on subsequent colonization in an equitable manner.

Future Goals for Model refinement

ArcINFO: The structure of these models is extremely versatile. Any number of coral species and propagules can be programmed to 'recruit and grow' within the framework of the existing model. Further refinement of the model in two areas would add precision to the model predictions: first, refine the density dependent aspect of propagule recruitment. If density of propagules are known, the number of recruits can be automatically based on the available area at the time of the recruitment event. A range of density values can also be randomly chosen. Second, apply actual vertical growth rates by species to the **Age Grid**. Species-specific vertical growth rates, once known, can be applied with each iteration to the **Age Grid**. This will create a more accurate representation of the vertical surface as the coral grows. Again, a target of mature coral dimension is required to produce a measure of percent recovery of coral structure.

SAS: At this point, the model requires significant modification to make it directly comparable to the ArcINFO version. First, we will install new propagule seeding events with each iteration of the model, as opposed to the single set as it occurs now. Second, we will assign different growth rates to a certain proportion of the random propagules to simulate multiple species colonization. Third, we will expand the iteration counter to make it able to track each pixel based on time of colonization and provide the capacity for simulating vertical growth. Finally, we will adapt the front end of the model to incorporate the same input data set as is used in the ArcINFO version.

For both modeling approaches (ArcINFO and SAS), prediction was only as good as the data that was put into it. More studies that examine species-specific propagule recruitment, horizontal and vertical growth within damaged coral reefs need to be employed. These data are necessary for the future refinement of the model and will aid in achieving accurate predictions for coral recovery. However, it is our opinion that given the assumptions of the model, it is exceedingly fair as it errs dramatically in favor of a Responsible Party.

Literature Cited

Connell J.H., Hughes T.P. and Wallace C.C. (1997) A 30-year study of coral abundance, recruitment and disturbance at several scales in space and time. Ecological Monongraphs. 67(4) 461-488.

Fonseca M.S., Julius B.E. and Kenworthy W.J. 2000. Integrating biology and economics in seagrass restoration: how much is enough? Ecological Engineering. 15:227-237

Fonseca, M.S. Whitfield P.E. Colby D.R., and Kenworthy W.J. In prep. Modeling the effect of injury geometry on the recovery of seagrasses. For submission to Restoration Ecology.

Gonzales E.K. (2000) Distinguishing between modes of dispersal by introduced eastern grey squirrels (Sciurus carolinensis). Master Thesis, University of Guelph,Canada http://www.uoguelph.ca/zoology/research/squirrel/

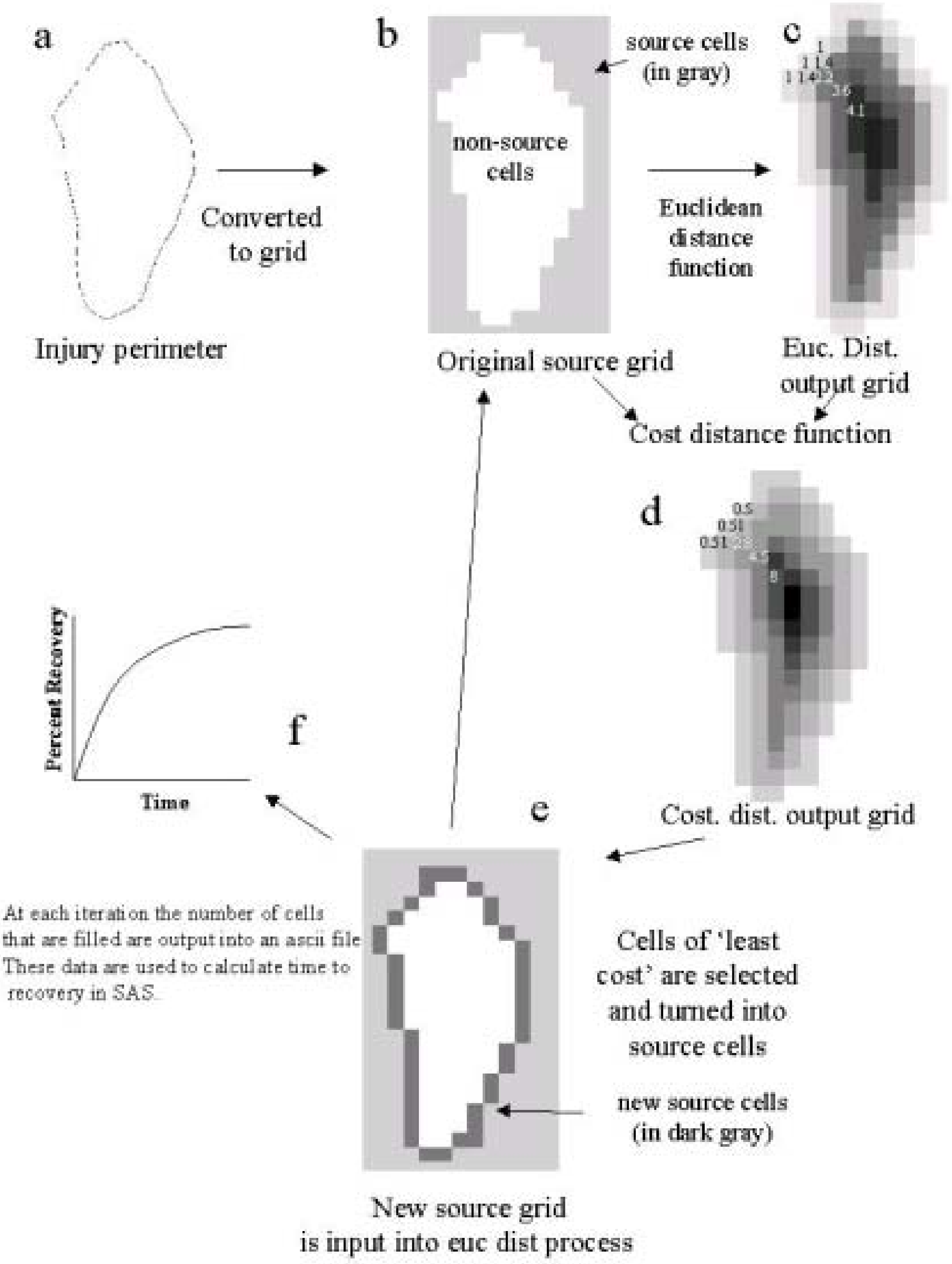
Halpin P.N and Bunn A.G. (2000) Using GIS to compute a least-cost distance matrix: A comparison of terrestrial and marine ecological applications. In: 20th Annual ESRI International Conference Proceedings. San Diego, CA. June26-30. http://www.esri.com/library/userconf/proc00/professional/papers/PAP890/p890.htm

Kenworthy W.J., Fonseca M.S., Whitfield P.E., Hammerstrom K.A.. In press. Analysis of seagrass recovery in experimental excavations and propeller scar disturbances in the Florida Keys National Marine Sanctuary. Journal of Coastal Research.

Strager J.M., Yuill C.B. Wood P.B. (2000) Landscape-based riparian habitat modeling for amphibians and reptiles using ARC/INFO GRID and ArcView GIS. In: 20th Annual ESRI International Conference Proceedings. San Diego, CA June 26 –30. http://www.esri.com/library/userconf/proc00/professional/papers/PAP575/p575.htm

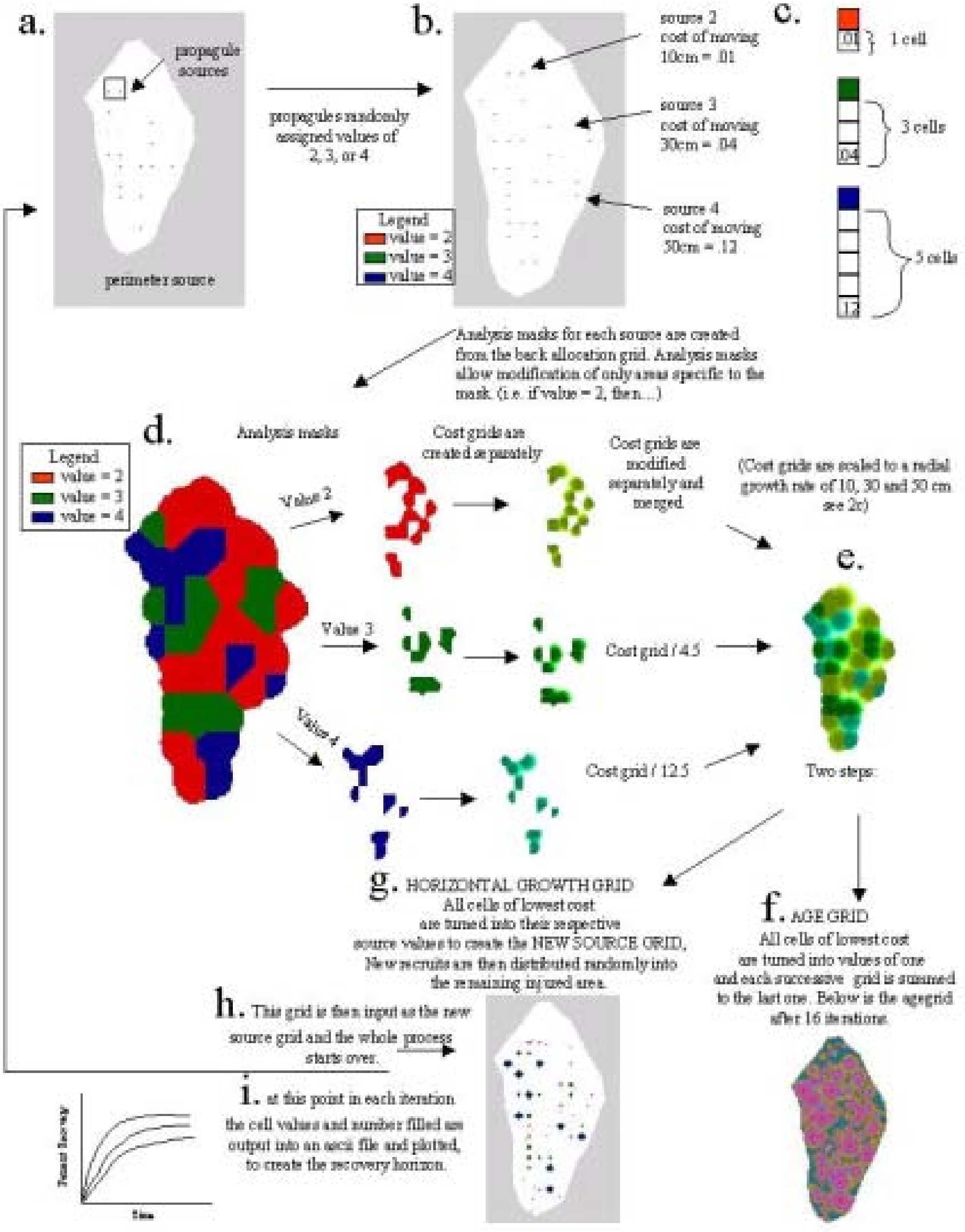
Zengel S. and Hinkeldey, H. (2001) Coral reef recovery: literature review and recommendations for damage assessment and restoration planning. NOAA Damage Assessment Center Contract # 50-DSNC-7-90032 to Industrial Economics, Incorporated, Cambridge, MA.

Figure 1. Seagrass Recovery Model process in Arc/INFO.



At each iteration the number of cells that are filled are output into an ascii file These data are used to calculate time to recovery in SAS.

Figure 2. Coral Modeling Process



Analysis masks for each source are created from the back allocation grid. Analysis masks allow modification of only areas specific to the mask. (i.e. if value = 2, then...)

(Cost grids are scaled to a radial growth rate of 10, 30 and 50 cm see 2c)

g. HORIZONTAL GROWTH GRID
All cells of lowest cost are turned into their respective source values to create the NEW SOURCE GRID, New recruits are then distributed randomly into the remaining injured area.

f. AGE GRID
All cells of lowest cost are turned into values of one and each successive grid is summed to the last one. Below is the agegrid after 16 iterations.

h. This grid is then input as the new source grid and the whole process starts over.

i. at this point in each iteration the cell values and number filled are output into an ascii file and plotted, to create the recovery horizon.

Figure 3. Top panel is the 2-dimensional illustration of the age grid. Bottom panel is a 3-dimensional representation of the same age grid.

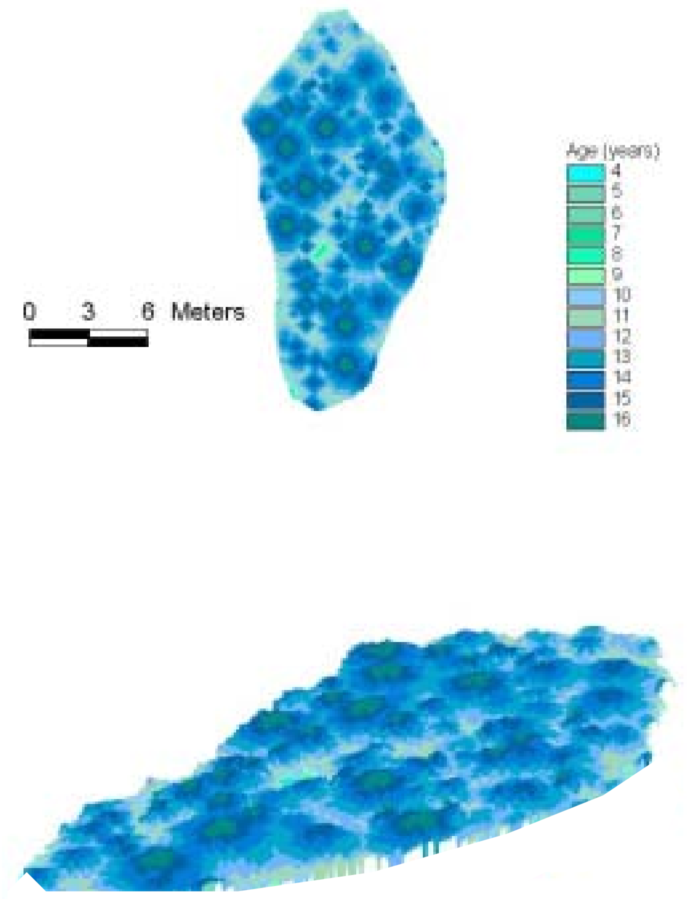


Figure 4. Example of SAS output based on accumulation of colonized cells, for each of the three coral species, over time. 'pctrec' is the percent recovery of each coral species, scaled to the area ultimately colonized by that species. A cumulative percent recovery based on all coral species' contribution to the injury area is also available for computing a net injury recovery trajectory.

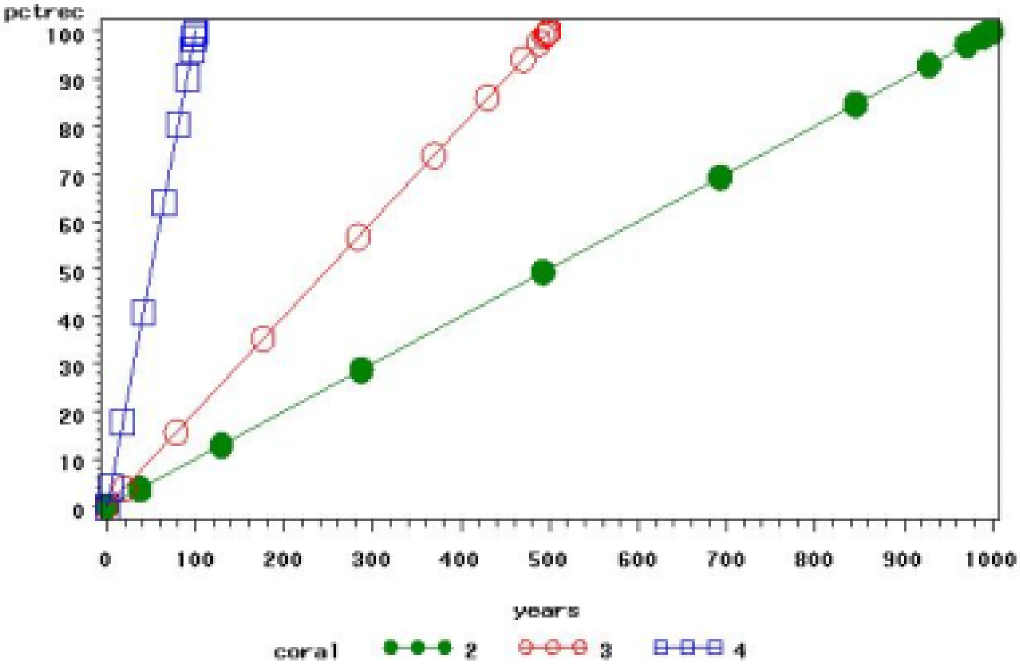


Figure 5. Examples of coral recovery from a single propagule rain using the SAS-based model; numbers refer to iterations of the model.